Ch 3. Functions and an Introduction to Recursion

Math Library Functions

Function	Description
`ceil( x )`	rounds *x* to the smallest integer not less than *x*
`cos( x )`	trigonometric cosine of *x* (*x* in radians)
`exp( x )`	exponential function e^x
`fabs( x )`	absolute value of *x*
`floor( x )`	rounds *x* to the largest integer not greater than *x*
`fmod( x, y )`	remainder of *x*/*y* as a floating-point number

Math Library Functions

Function	Description
`log( x )`	natural logarithm of *x* (base *e*)
`log10( x )`	logarithm of *x* (base 10)
`pow( x, y )`	*x* raised to power *y* (x^y)
`sin( x )`	trigonometric sine of *x* (*x* in radians)
`sqrt( x )`	square root of *x* (where *x* is a nonnegative value)
`tan( x )`	trigonometric tangent of *x* (*x* in radians)

Pass by Reference

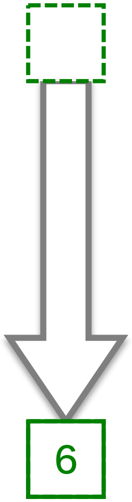
```
int main()
{
   int number = 6;

   cout << "number: " << number << endl;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
}
```

Pass by Reference

```
void passByReference( int &ref )
{
   ref = 8;
}
```

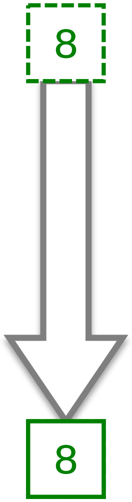


Output

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}
```

Pass by Reference

```
void passByReference( int &ref )
{
   ref = 8;
}
```

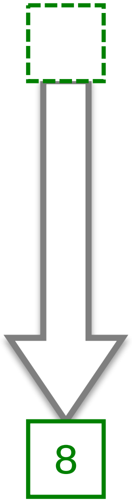


Output

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}
```

Pass by Reference

```
void passByReference( int &ref )
{
   ref = 8;
}
```



```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}
```

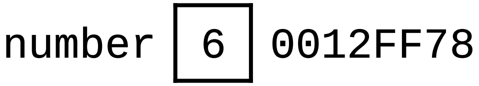
Output

```
number: 8
```

Pass by Reference

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   ref = 8;
}
```



Output

Pass by Reference

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   ref = 8;
}
```

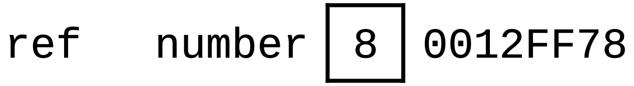
ref number 6 0012FF78

Output

Pass by Reference

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   ref = 8;
}
```

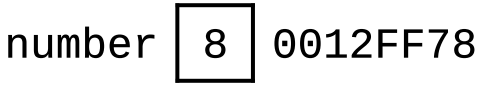


Output

Pass by Reference

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   ref = 8;
}
```


number 8 0012FF78

Output

Pass by Reference

```
int main()
{
   int number = 6;
   passByReference( number );
   cout << "number: " << number << endl;
}

void passByReference( int &ref )
{
   ref = 8;
}
```

number 8 0012FF78

Output

```
number: 8
```

Reference

```cpp
int main()
{
   int number = 6;
   int &ref = number; // ref refers to (is an alias for) number

   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

```
number: 6
ref: 6
number: 8
ref: 8
```

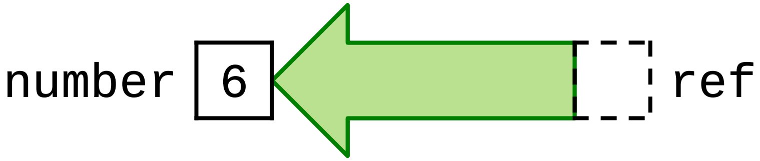
Reference

number 6

Output

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

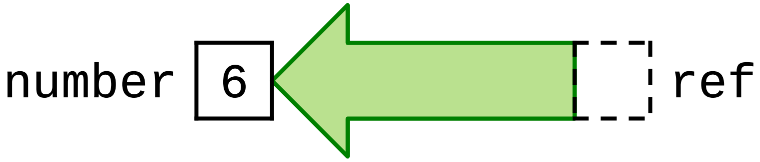


number 6 ref

Output

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

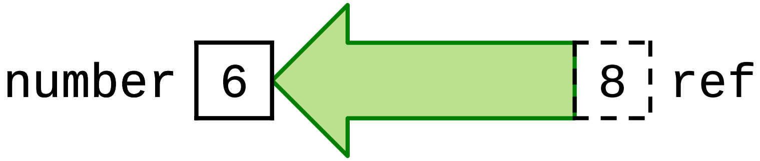


Output

```
number: 6
ref: 6
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

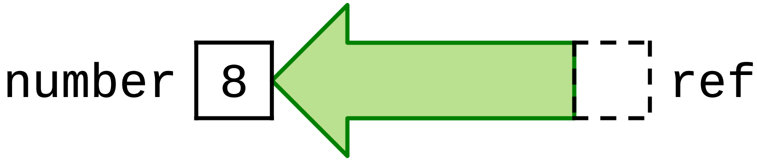


Output

```
number: 6
ref: 6
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

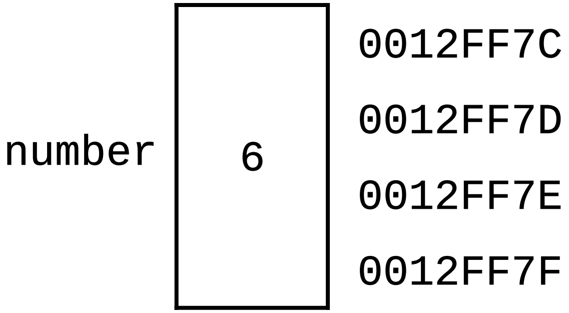


Output

```
number: 6
ref: 6
ref: 8
number: 8
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

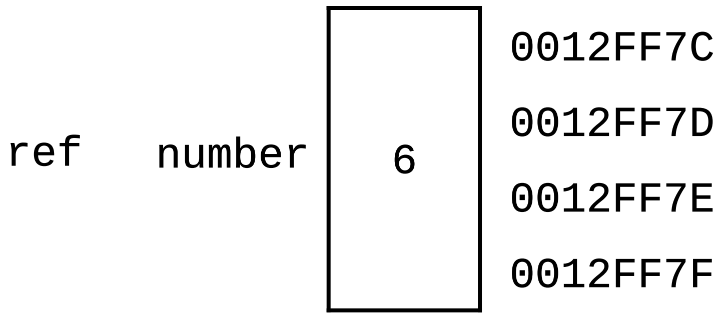
Reference



Output

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

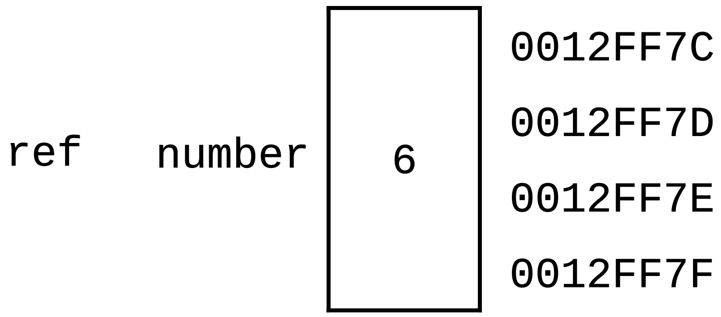
Reference



Output

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

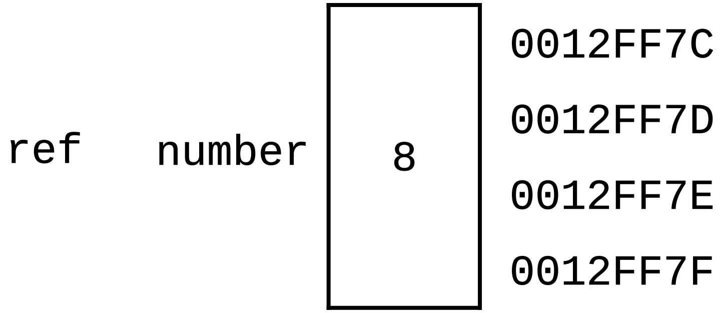


Output

```
number: 6
ref: 6
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference

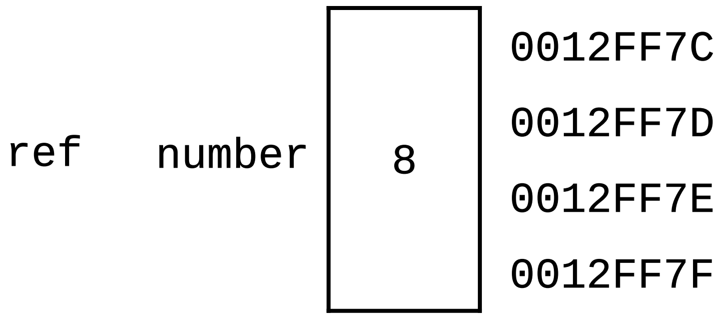


Output

```
number: 6
ref: 6
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

Reference



Output

```
number: 6
ref: 6
ref: 8
number: 8
```

```
int main()
{
   int number = 6;
   int &ref = number;
   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

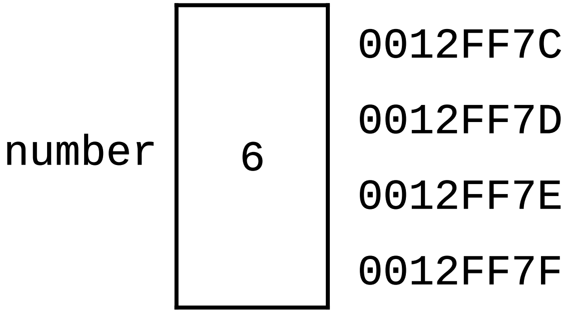
Reference

```
int main()
{
   int number = 6;
   int &ref; // Error: ref must be initialized

   cout << "number: " << number << endl;
   cout << "ref: " << ref << endl;
   ref = 8;
   cout << "ref: " << ref << endl;
   cout << "number: " << number << endl;
}
```

```
error C2530: 'ref': 參考必須初始化
```

Comparison

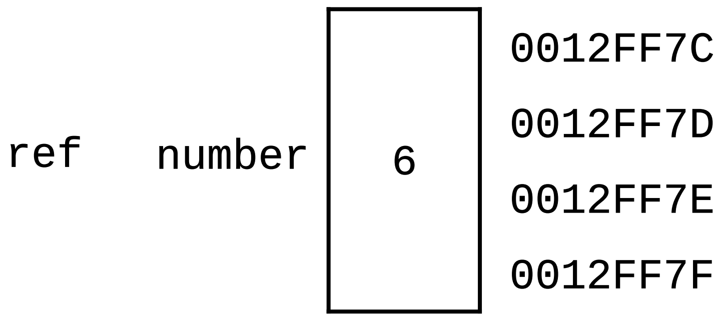


```
int main()
{
   int number = 6;
   int &ref = number;
   ref = 8;
}
```

```
int main()
{
   int number = 6;
   passByReference( number );
}

void passByReference( int &ref )
{
   ref = 8;
}
```

Comparison



ref

number

6

0012FF7C

0012FF7D

0012FF7E

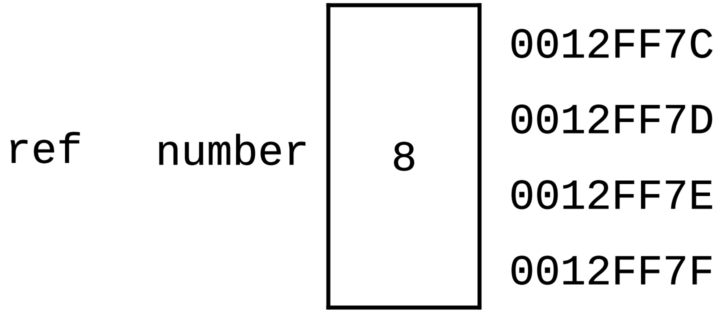
0012FF7F

```
int main()
{
   int number = 6;
   int &ref = number;
   ref = 8;
}
```

```
int main()
{
   int number = 6;
   passByReference( number );
}

void passByReference( int &ref )
{
   ref = 8;
}
```

Comparison



ref number 8

0012FF7C
0012FF7D
0012FF7E
0012FF7F

```
int main()
{
   int number = 6;
   int &ref = number;
   ref = 8;
}
```

```
int main()
{
   int number = 6;
   passByReference( number );
}

void passByReference( int &ref )
{
   ref = 8;
}
```

```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl;

   // pass array a to passArray by reference
   passArray( a, size );

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl << endl;

   cout << "a[ 2 ]: " << a[ 2 ] << endl;
   passElement( a[ 2 ] ); // pass array element a[ 2 ] by value
   cout << "a[ 2 ]: " << a[ 2 ] << endl << endl;
}
```

```
void passArray( int b[], int size )
{
   for( int i = 0; i < size; i++ )
      b[ i ] *= 2;
}

void passElement( int element )
{
   element *= 2;
}
```

Output

```
0   1   2   3   4
0   2   4   6   8

a[ 2 ]: 4
a[ 2 ]: 4
```

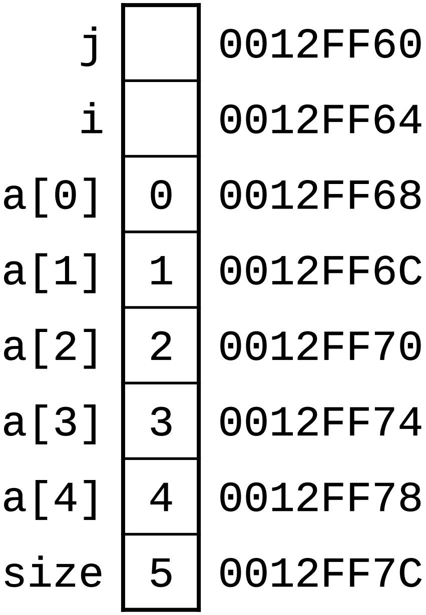
```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl;

   passArray( a, size );

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl << endl;
}

void passArray( int b[], int size )
{
   for( int i = 0; i < size; i++ )
      b[ i ] *= 2;
}
```



j		0012FF60
i		0012FF64
a[0]	0	0012FF68
a[1]	1	0012FF6C
a[2]	2	0012FF70
a[3]	3	0012FF74
a[4]	4	0012FF78
size	5	0012FF7C

```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl;

   passArray( a, size );

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl << endl;
}

void passArray( int b[], int size )
{
   for( int i = 0; i < size; i++ )
      b[ i ] *= 2;
}
```

	j		0012FF60
	i		0012FF64
b[0]	a[0]	0	0012FF68
b[1]	a[1]	1	0012FF6C
b[2]	a[2]	2	0012FF70
b[3]	a[3]	3	0012FF74
b[4]	a[4]	4	0012FF78
	size	5	0012FF7C

i		0012FEF0
size	5	0012FF00

```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl;

   passArray( a, size );

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl << endl;
}

void passArray( int b[], int size )
{
   for( int i = 0; i < size; i++ )
      b[ i ] *= 2;
}
```

	j		0012FF60
	i		0012FF64
b[0]	a[0]	0	0012FF68
b[1]	a[1]	2	0012FF6C
b[2]	a[2]	4	0012FF70
b[3]	a[3]	6	0012FF74
b[4]	a[4]	8	0012FF78
	size	5	0012FF7C

i		0012FEF0
size	5	0012FF00

```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl;

   passArray( a, size );

   for( int i = 0; i < size; i++ )
      cout << a[ i ] << "  ";
   cout << endl << endl;
}

void passArray( int b[], int size )
{
   for( int i = 0; i < size; i++ )
      b[ i ] *= 2;
}
```

j		0012FF60
i		0012FF64
a[0]	0	0012FF68
a[1]	2	0012FF6C
a[2]	4	0012FF70
a[3]	6	0012FF74
a[4]	8	0012FF78
size	5	0012FF7C

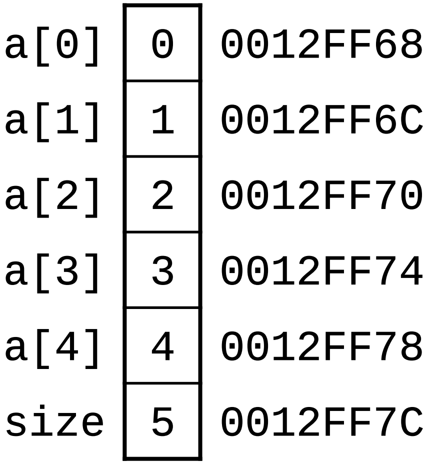
```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   cout << "a[ 2 ]: " << a[ 2 ] << endl;

   passElement( a[ 2 ] );

   cout << "a[ 2 ]: " << a[ 2 ] << endl;
}

void passElement( int element )
{
   element *= 2;
}
```



a[0]	0	0012FF68
a[1]	1	0012FF6C
a[2]	2	0012FF70
a[3]	3	0012FF74
a[4]	4	0012FF78
size	5	0012FF7C

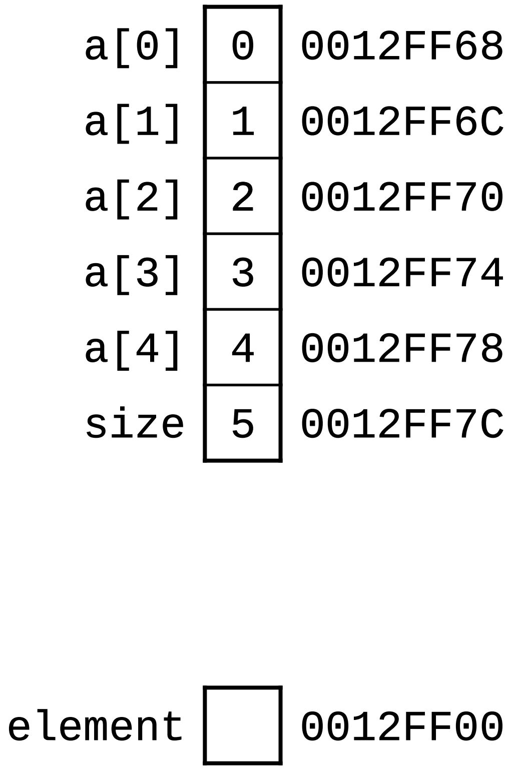
```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   cout << "a[ 2 ]: " << a[ 2 ] << endl;

   passElement( a[ 2 ] );

   cout << "a[ 2 ]: " << a[ 2 ] << endl;
}

void passElement( int element )
{
   element *= 2;
}
```



a[0]	0	0012FF68
a[1]	1	0012FF6C
a[2]	2	0012FF70
a[3]	3	0012FF74
a[4]	4	0012FF78
size	5	0012FF7C
element		0012FF00

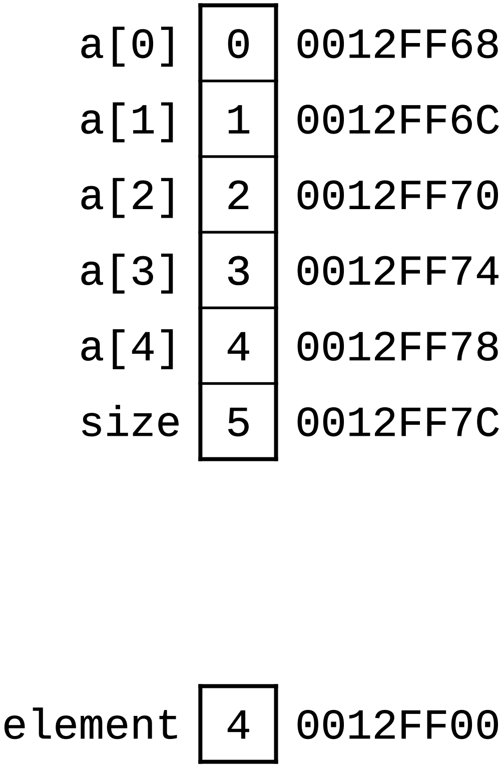
```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   cout << "a[ 2 ]: " << a[ 2 ] << endl;

   passElement( a[ 2 ] );

   cout << "a[ 2 ]: " << a[ 2 ] << endl;
}

void passElement( int element )
{
   element *= 2;
}
```



a[0]	0	0012FF68
a[1]	1	0012FF6C
a[2]	2	0012FF70
a[3]	3	0012FF74
a[4]	4	0012FF78
size	5	0012FF7C

element	4	0012FF00

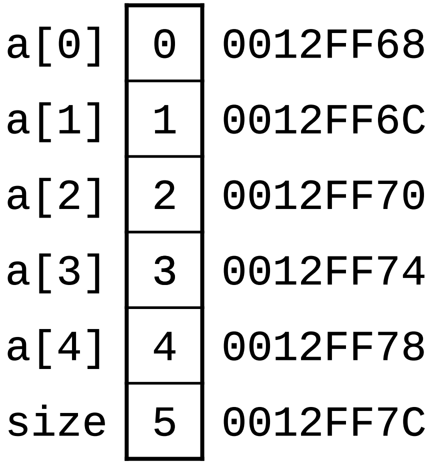
```
int main()
{
   const int size = 5;
   int a[ size ] = { 0, 1, 2, 3, 4 };

   cout << "a[ 2 ]: " << a[ 2 ] << endl;

   passElement( a[ 2 ] );

   cout << "a[ 2 ]: " << a[ 2 ] << endl;
}

void passElement( int element )
{
   element *= 2;
}
```



a[0]	0	0012FF68
a[1]	1	0012FF6C
a[2]	2	0012FF70
a[3]	3	0012FF74
a[4]	4	0012FF78
size	5	0012FF7C

Random Number Generator

```cpp
#include <iostream>
using namespace std;

int main()
{
   for( int i{ 1 }; i <= 20; ++i )
   {
      cout << 1 + rand() % 6 << "  ";

      if( i % 5 == 0 )
         cout << endl;
   }
   cout << endl;
}
```

Output

```
6  6  5  5  6
5  1  1  5  3
6  6  2  4  2
6  2  3  4  1
```

srand

```
int main()
{
   unsigned int seed;
   cout << "Enter seed: ";
   cin >> seed;
   srand( seed );

   for( int i{ 1 }; i <= 20; ++i )
   {
      cout << 1 + rand() % 6 << "  ";

      if( i % 5 == 0 )
         cout << endl;
   }
   cout << endl;
}
```

```
Enter seed: 1
  6  6  5  5  6
  5  1  1  5  3
  6  6  2  4  2
  6  2  3  4  1
```

```
Enter seed: 2
  4  3  1  6  1
  1  5  2  2  5
  2  2  5  1  6
  4  6  2  5  6
```

```
Enter seed: 3
  1  3  1  2  6
  4  3  2  2  1
  4  4  2  5  3
  6  2  6  6  3
```

```
#include <iostream>
using namespace std;

int main()
{
   cout << "Current time: " << time( 0 ) << endl << endl;

   srand( static_cast< unsigned int >( time( 0 ) ) );

   for( int i{ 1 }; i <= 20; ++i )
   {
      cout << 1 + rand() % 6 << "  ";

      if( i % 5 == 0 )
         cout << endl;
   }
   cout << endl;
}
```

Output

```
Current time: 1693880338

2  6  3  1  1
1  4  6  2  1
5  6  6  2  1
4  6  3  3  2
```

```
unsigned int randState = 1;

// Seeds the random number generator with the provided integer.
void srand( unsigned int const seed )
{
   randState = seed;
}

// Returns a pseudorandom number in the range [0,32767].
int rand()
{
   const unsigned int randMax = 32767; // 0x7fff

   randState = randState * 214013 + 2531011;

   return ( randState >> 16 ) & randMax;
   return ( randState / 65536 ) & randMax;
   return ( randState / 65536 ) % ( randMax + 1 );
}
```

Inline Functions

```
#include <iostream>
using namespace std;

inline int square( const int sideLength )
{
   return sideLength * sideLength;
}

int main()
{
   int side;
   cout << "Enter the side length of your square: ";
   cin >> side;

   cout << "Area of square with side "
        << side << " is " << square( side ) << endl;
}
```

Inline Functions

```
#include <iostream>
using namespace std;

int main()
{
   int side;
   cout << "Enter the side length of your square: ";
   cin >> side;

   cout << "Area of square with side "
        << side << " is " << side * side * side << endl;
}
```

```cpp
#include <iostream>
using namespace std;

inline int maximum( int a, int b, int c )
{
   int max = a;
   if ( b > max )
      max = b;
   if ( c > max )
      max = c;
   return max;
}

int main()
{
   int number1;
   int number2;
   int number3;

   cout << "Enter three integers: ";
   cin >> number1 >> number2 >> number3;

   cout << maximum( number1, number2, number3 ) << endl;
}
```

```cpp
#include <iostream>
using namespace std;

int main()
{
   int number1;
   int number2;
   int number3;

   cout << "Enter three integers: ";
   cin >> number1 >> number2 >> number3;

   int max = number1;
   if ( number2 > max )
      max = number2;
   if ( number3 > max )
      max = number3;
   cout << "Maximum is: "
        << max << endl;
}
```

Default Arguments

```cpp
#include <iostream>
using namespace std;

int area( int length = 1, int width = 1 );

int main()
{
   cout << "The default rectangle area is: " << area() << endl;

   cout << "The area of a rectangle with length 10 and width 1 is: "
        << area( 10 ) << endl;

   cout << "The area of a rectangle with length 10 and width 5 is: "
        << area( 10, 5 ) << endl;
}

int area( int length, int width )
{
   return length * width;
}
```

```
The default rectangle area is: 1
The area of a rectangle with length 10 and width 1 is: 10
The area of a rectangle with length 10 and width 5 is: 50
```

Unary Scope Resolution Operator

```cpp
#include <iostream>
using namespace std;

int n = 3;

int main()
{
   int n = 7;

   cout << "The value of local variable n = " << n << endl;

}
```

```
The value of local variable n = 7
The value of global variable n = 3
```

Unary Scope Resolution Operator

```
#include <iostream>
using namespace std;

int n = 3;

int main()
{
   int n = 7;

   cout << "The value of local variable n = " << n << endl;
   cout << "The value of global variable n = " << ::n << endl;
}
```

```
The value of local variable n = 7
The value of global variable n = 3
```

Function Overloading

```
int myAbs( int i )
{
  cout << "absolute value of int " << i << " is ";

  if( i < 0 )
    return -i;
  else
    return i;
}

float myAbs( float f )
{
  cout << "absolute value of float " << f << " is ";

  if( f < 0 )
    return -f;
  else
    return f;
}
```

Function Overloading

```
double myAbs( double d )
{
   cout << "absolute value of double " << d << " is ";

   if( d < 0 )
      return -d;
   else
      return d;
}

int main()
{
   int i = -3;
   float f = static_cast< float >( -3.3 );
   double d = -3.3;

   cout << myAbs( i ) << endl;
   cout << myAbs( f ) << endl;
   cout << myAbs( d ) << endl;
}
```

```
absolute value of int -3 is 3
absolute value of float -3.3 is 3.3
absolute value of double -3.3 is 3.3
```

Function Templates

```
template < typename T >
T myAbs( T number )
{
   cout << "absolute value of " << number << " is ";

   if( number < 0 )
      return -number;
   else
      return number;
}

int main()
{
   int i = -3;
   float f = static_cast< float >( -3.3 );
   double d = -3.3;

   cout << myAbs( i ) << endl;
   cout << myAbs( f ) << endl;
   cout << myAbs( d ) << endl;
}
```

```
absolute value of -3 is 3
absolute value of -3.3 is 3.3
absolute value of -3.3 is 3.3
```

```
template < typename T >
T myAbs( T number )
{
   cout << "absolute value of " << number << " is ";

   if( number < 0 )
      return -number;
   else
      return number;
}

int myAbs( int number )
{
   cout << "absolute value of " << number << " is ";

   if( number < 0 )
      return -number;
   else
      return number;
}
```

```
float myAbs( float number )
{
   cout << "absolute value of " << number << " is ";

   if( number < 0 )
      return -number;
   else
      return number;
}

double myAbs( double number )
{
   cout << "absolute value of " << number << " is ";

   if( number < 0 )
      return -number;
   else
      return number;
}
```